REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-98-

ß

0744

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. RI
	None Shown	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Visualization in Computational Fluid Dynamics: A Case Study	

6. AUTHORS
Robert Haimes and Dave Darmofal

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Massachusetts Institute of Technology	

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFOSR/NI 4040 Fairfax Dr, Suite 500 Arlington, VA 22203-1613	

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for Public Release	

13. ABSTRACT *(Maximum 200 words)*
No Abstract Provided

14. SUBJECT TERMS	15. NUMBER OF PAGES
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

DTIC QUALITY INSPECTED 3

Visualization in Computational Fluid Dynamics : A Case Study

Robert Haimes and Dave Darmofal

Computational Fluid Dynamics Laboratory

Department of Aeronautics and Astronautics

Massachusetts Institute of Technology, Cambridge, MA 02139

1 Introduction

The data produced by Computational Fluid Dynamics (CFD) codes creates many problems for visualization systems. The data is volumetric; in other words, the information of interest is throughout a volume and not just on a surface. This volume may be broken up in many different ways. The simplest form of discretization is to cut the volume into a structured set of hexahedra. On the other extreme the volume may be broken into tetrahedra in an irregular manner producing an "unstructured" grid. CFD applications can also produce very large data-sets. For example, it is not uncommon to perform a full airplane calculation with over a million nodes. Also, researchers are now looking at 3D CFD calculations that are time varying! Traditional visualization techniques based on CAD output only provide surface information and cannot give the CFD investigator much help in quickly understanding the results of these codes.

Within the last couple of years graphics workstations have advanced to the point where the high-end workstations, due to their speed in both "number-crunching" and 3D drawing capabilities, are appropriate for interactive volumetric scientific visualization. VISUAL3[1][2] was written to take advantage of these workstations in tackling the problems of investigating the results from CFD calculations.

2 Motivation for Creating VISUAL3

The work on VISUAL3 began because there was no hardware/software combination on the market that would satisify the post-processing needs of the CFD Lab at MIT. The baseline requirements for VISUAL3 were the ability to handle both structured and unstructured grids, the display of unsteady data, the ease of use by students, and above all high interactivity.

During Computer-Aided Education software development for undergraduates at M.I.T.'s Project Athena it became clear the interactivity was of paramount importance in conveying information to the user. Having to wait for results of a user action breaks the train of thought and can be frustrating. In order for a visualization package to be a fruitful tool it must be able to answer questions with a minimum delay in time.

At M.I.T., we have the luxury of not having to sell our software in order to survive. Therefore, VISUAL3 was first targetted at high-end hardware to satisify the Lab's needs and not the lowest capability hardware in a manufacturer's line. This ultimately allowed research into new techniques while maintaining high performance and interactivity. For example, a new efficient method was developed for cutting through the volume (either iso-surface or planar cut) that does not require voxels or "marching cubes" in that neither is appropriate for unstructured meshes[3].

In general, the scientific visualization software available seems to be more aimed at generating pictures for presentation than at understanding of fluid dynamics. An important design goal for VISUAL3 was that it be a tool in the investigation and interpretation of computational results. When this goal is met, a side benefit is that with better viewing and understanding of the flow results, the visualization software can be used for flow solver debugging. For example, unlike packages that generate "pretty" pictures, VISUAL3 performs all shading in a facetted manner. This is part of a design philosophy to give the investigator only the data. Phong shading, or anything that may smooth out blemishes (techniques routinely used in CAD) are always avoided.

3 Immediate Mode

The only 3D software standard currently available is PHIGS. Using PHIGS for VISUAL3 development would have had the obvious advantage of portability but PHIGS has some significant drawbacks. Many PHIGS implementations support only one window, but VISUAL3's design requires a multi-window approach. PHIGS is designed for CAD applications where the complete object is pre-defined and any movements or changes to the object are done by local transforms. This allows one part of the object to move in respect to another. In VISUAL3, using scanning tools, surfaces are always being created and destroyed. In an unsteady application all the data changes from one time to the next. In fact, changing the scalar color mapping in PHIGS requires that the whole object be rebuilt since color values are a part of the PHIGS data structure. Furthermore, most aspects of CFD visualization do not rely upon

an object-oriented rendering which is one of the advantages of a PHIGS style standard. As a result of these problems, PHIGS was rejected for the implementation of VISUAL3.

VISUAL3 was developed instead using Stardent's XFDI extension to the X window system. XFDI provides immediate mode 3D support enhancements to X. Therefore most of VISUAL3 is an X application with a few non-standard internal calls to perform Gouraud shading for surfaces and 3D support for drawing lines and triangles with hidden surface removal. This is very much the same model as the current PEX standard. It is unfortunate that the only standard Application Programming Interface (API) into PEX is currently PHIGS. VISUAL3 has been ported to a DECstation 5000 using DECs PEXlib (a library of bindings that build PEX packets to send to the server analogous to Xlib for straight X servers). It is most important that some acceptable alternative API that supports immediate mode applications be agreed upon so that packages that perform scientific visualization can have the same kind of portability as other software!

4 Multiple Windows

4.1 A Dimensional Approach

An extremely important aspect of CFD visualization is the ability to simultaneously view data in multiple ways. Thus, a multi-windowing system is an essential part of VISUAL3's design. In particular, the ability to look at data in one-, two-, and three-dimensions simultaneously is extremely important. Many visualization software packages offer the ability to view an object from different positions simultaneously. We have found that rotation and translation of an object can fulfill the same purposes and reduce screen clutter. VISUAL3's window set-up (see Figure 2 for layout) is designed with a dimensional approach in mind. The three main plotting windows:

- 3D Window

  The 3D window displays data on three-dimensional surfaces, either from the bounding domain or from cuts and iso-surfaces. It also displays three-dimensional lines such as tufts and streamlines. The objects in the 3D window can be rotated, translated and enlarged using the dialbox.

- 2D Window

  The 2D window is used to display data on a mapped domain or dynamic surface (for which there exists a mapping to an (x', y') coordinate system). The user can independently adjust the view in this window.

- 1D Window

  The 1D window is used to display one dimensional data which is generated by various functions in the 2D window or from mapping streamlines.

The reduction of dimensions from each window can significantly aid in the understanding of a flow field. Often, patterns are much easier to recognize on two-dimensional planar cuts through a domain than in three-dimensions. Furthermore, information on a two-dimensional surface may be reduced to one dimension. For example, the user may interactively set a line in the two-dimensional window and a plot of the current scalar will be displayed in the one-dimensional window.

4.2 Cursor Mapping

Another important concept related to the dimensional windowing approach is cursor mapping. Cursor mapping is the simultaneous display of the current cursor position in the one-, two-, and three-dimensional views. For example, suppose a line plot is displayed in the 1D window. When pointing (with the cursor) in the 1D window, the user is actually pointing in 2-space by reversing the 2D to 1D mapping. Similarly, this point in 2-space corresponds to a point in 3-space. This is a simple and powerful concept. This cursor mapping is extremely useful with streamlines. The calculation of a streamline requires an initial 3-space location to begin numerical integration. With structured grids, this is often done by simply selecting a specific computational grid line or node. However, with unstructured grids, this technique is impossible. With cursor mapping, the user simply points to a location in the two-dimensional window which corresponds to a unique 3-space point from which a streamline may be spawned. Without a multi-window system pointing in 3D is, at best, difficult.

5 Visualization Tools

Besides multiple windows, we believe that having multiple visualization tools is also extremely important. Multiple tools allow the user to view the same data in many ways, often providing additional insight. In general, these tools have been stimulated by user feedback. Consequently, VISUAL3 is constantly adding new capabilities. Although many of the techniques are extremely simple to implement, they are often the most widely-used tools.

Visualization tools may be divided hierarchically according to their abilities[4]. The main tasks during visualization of a CFD data set are feature identification, scanning, and probing. Identification techniques help the user to locate flow features over the entire domain and can save a great deal of effort and time during visualization. Scanning techniques allow the user to interactively search the domain by varying a single parameter. For example, a cutting plane, a two-dimensional surface slice through the domain, is a scanning technique because its position may be changed along its normal. Thus, the plane may be scanned through the domain. Finally, probing techniques are very localized visualization tools. A point probe which returns the scalar value from the current cursor position is a probing technique. This breakdown of visualization methods is in some sense sim-

ply a reduction of dimensions. In other words, identification techniques are three-dimensional, scanning techniques are two-dimensional, and probing are one-dimensional. As with multiple dimensional windows, we again stress the importance of reduction of dimensions as an integral part to effective and efficient visualization. The remainder of this section briefly describes some of the various tools we have implemented in VISUAL3. A more complete description of the tools and algorithm issues may be found in [1][2][4].

5.1 Identification Tools

Probably the weakest link in visualization is the ability to quickly identify features. Although some work has been done, a great deal of improvement can still be made in useful feature identification algorithms. Currently, VISUAL3 supports vector clouds for feature identification purposes. Vector clouds display the local velocity field in areas which have been thresholded to display only the interesting portions of the flow. In the vortical flow over a delta wing, total pressure loss is high in the vortices; thus, thresholding away the low total pressure regions should result in an image showing the vortical regions of the flow. Figure 1 is an example of vector clouds using total pressure loss as the threshold variable in a transonic delta wing flow[5]. The primary and secondary vortices can be cleary seen in the image.

We also are researching other possible feature extraction methods. For example, a discontinuity locater is being developed to locate the positions of shocks within a flow field. See Section 5.2.2 for more on shock detection. Recently, an X-ray feature identification algorithm was also implemented[6]. Rays are sent through a computational domain and the ray intensity is decreased according to the scalar values it travels through. However, the X-ray technique requires essentially a ray tracing algorithm which is usually slow and hampers interactivity.

5.2 Scanning Tools

Scanning techniques, although still relatively recent developments, are much more established than feature identification techniques. This set of methods allows the researcher to scan through the domain either through space or through scalar values. We discuss some of the more widely used scanning techniques below.

5.2.1 Cutting Planes

Cutting planes are planar surfaces which slice through the computational domain. The scalar values at the nodes of the cut cells are interpolated onto the planar surface and then rendered in two-dimensions. The plane may then be scanned in its normal direction allowing the user to investigate the entire domain if necessary. The plane normal direction, location, and size may be interactively set by inputs to the dialbox. A very efficient algorithm for unstructured grid

cutting planes has been successfully implemented by Giles and Haimes[1]. We again use the transonic delta wing data set show the result of a cutting plane in all windows. The line probe can also be seen to be active in the 1D window.

5.2.2 Iso-Surfaces

Iso-surfaces are three-dimensional surfaces of a constant scalar value. In reality, cutting planes are simply the geometric subset of iso-surfaces where the constant value parameter is a geometric coordinate. Iso-surfaces are extremely useful for two purposes. First, the scanning of iso-surfaces is an effective method for visualizing three-dimensional scalar gradient fields since each consecutive surface represents a small change in a scalar from the previous surface. Thus, in areas where gradients are locally high, consecutive iso-surfaces will essentially maintain the same shape. Similarly, in areas where the gradients are low, consecutive iso-surfaces will displace considerably. Second, iso-surfaces may be useful as a feature identification technique. If a flow feature may be identified by unique scalar value, then an iso-surface of that value will show the desired feature. A shock finding algorithm has been implemented by constructing an iso-surface where the Mach number normal to the shock surface is 1.0[4]. Figure 3 shows the shock surfaces in a hypersonic delta wing calculation by Lee[7].

5.2.3 Tufts

Tufts are small vectors whose direction and size represent the vector field, which is typically the velocity vector. A regular grid of points on a cutting plane defines where the tufts are displayed. The projection of the tufts appears on the cutting plane. The three-dimensional view shows the tufts in their true position. A slight variation places tufts only at the intersection of the cutting plane and a computational cell edge. This option limits errors arising from interpolation. Since tufts are used on cutting planes which may be scanned, they have been defined as a scanning technique also. The tufts in Figure 4 can be seen in both the two-dimensional and three-dimensional windows. The tufts highlight the vortical nature of the flow around the wing.

5.3 Probing Tools

Probing methods, the final type of visualization technique, provide the most localized information. These tools are primarily used in the final step of investigating a flow feature to gather quantitative information such as the pressure distribution on a surface, or velocity profiles in boundary layers. Typically, these probes mimic the action of common experimental techniques and, therefore, are often used in comparing calculations to experiments. We illustrate just a few of these probes below.

3

5.3.1 Pathlines

A pathline is a line through the flow field which is everywhere tangent to the local vector field. If the vector field is the velocity field, then the pathline is termed a streamline. In a steady flow, a streamline is the path a particle follows when released in the flow. The results from three streamlines may be seen in Figure 5.

5.3.2 Tubes

Tubes are pathlines with circular cross-sectional areas based upon the local crossflow divergence. Thus, the thickness of streamtube cross-sections represents the local expansion of the flow field. Besides providing information about the local expansion of the flow, tubes are much easier to understand than regular pathlines because of the additional thickness. The intermixing of tubes can be visualized; with pathlines, the relative positions of several lines is hard to determine from a three-dimensional view. The streamtubes in Figure 5 are the same streamlines from above.

5.3.3 Point Probe

The point probe has two major functions: the local scalar value is displayed in a text window, and the local vector is projected on the two-dimensional plane. As a result of the cursor mapping function discussed earlier, this also displays the local vector in the three-dimensional window.

5.3.4 Boundary Layer Probe

The boundary layer probe plots the variation of a scalar along a direction normal to a boundary surface. The normal may be interactively positioned anywhere on the boundary surface by moving the cursor in the 2-D window. As the name indicates, this probe is effective for, investigating boundary layer effects.

5.3.5 Pathline Probe

The pathline probe displays the variation of a scalar along a pathline. The cursor mapping ability allows the pathline probe to pinpoint the exact location of unusual pathwise behaviors. An additional feature is the ability to use the pathline probe to position a cutting plane. When the probe is on, the cutting plane may be placed at any location on the current pathline with the plane normal being the local path-direction at the cursor location. Using this tool, a cutting plane can be easily placed at a shock or any other interesting feature along the pathline.

6 Conclusions

The lack of adequate scientific visualization software for volumetric data has led us to the in-house development of VISUAL3 – an unstructured, unsteady, three-dimensional visualization package. During the design, careful attention was paid to high interactivity and the understanding of fluid dynamics rather than the production of high-quality pictures. The use of multiple windows as a reduction of dimensions is very helpful in the comprehension of results. VISUAL3 also supports a large variety of visualization tools. In particular, visualization tools which automatically locate flow features could be significantly improved. These identification techniques can save large amounts of time by detecting flow features without the need to painstakingly scan an entire domain.

References

[1] M. Giles and R. Haimes. Advanced Interactive Visualization for CFD. *Computing Systems in Engineering*, 1(1):51–62, 1990.

[2] R. Haimes and M. Giles. VISUAL3: Interactive Unsteady Unstructured 3D Visualization. AIAA Paper 91-0794, 1991.

[3] W. E. Lorensen and H. E. Cline. Marching Cubes in a High Resolution 3D Surface Construction Algorithm. *ACM Computer Graphics*, Vol. 21(4):pp. 163–169, 1987.

[4] D. Darmofal. Hierarchal Visualization of Three-Dimensional Vortical Flow Calculations. Master's thesis, M.I.T., March 1991.

[5] T. Becker. Hybrid 3-D Euler/Navier-Stokes Calculations of Transonic Vortex Flows over the NTF Delta Wing. Master's thesis, M.I.T., August 1989.

[6] D. Darmofal and R. Haimes. Visual Feature Identification for 3-D Data Sets. AIAA Paper 91-1583, 1991.

[7] K.-M. Lee. Numerical Simulation of Hypersonic Flow Over a Blunt Leading Edge Delta Wing. Master's thesis, M.I.T., October 1989.
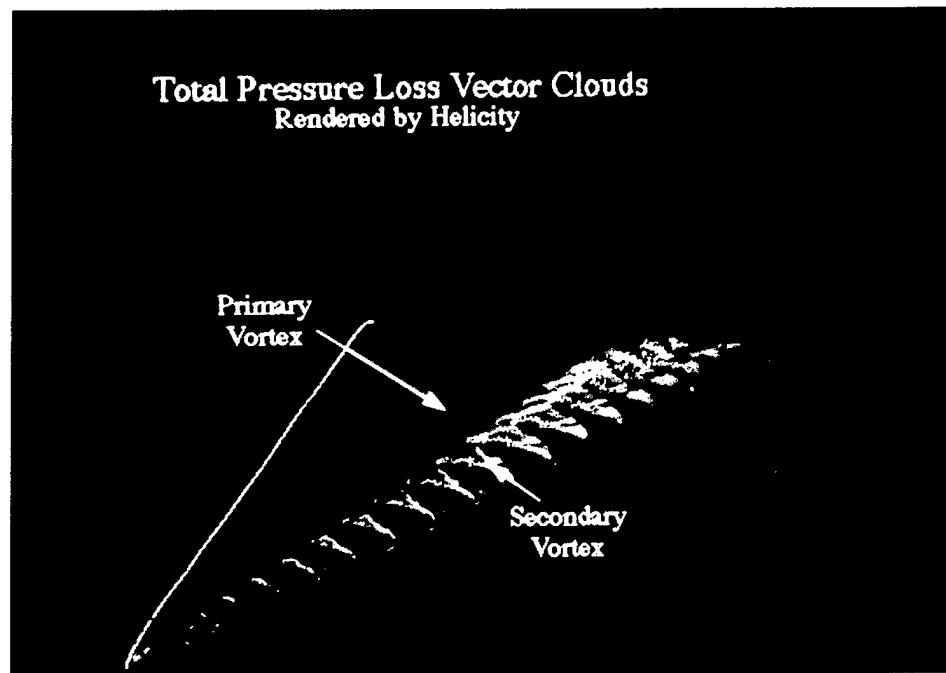
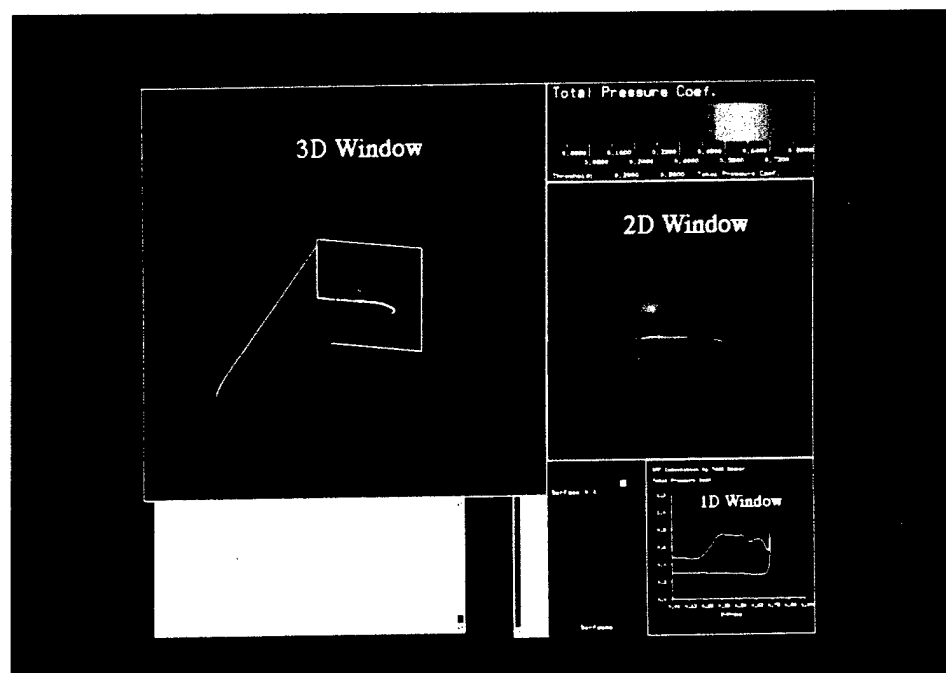Figure 1: Total Pressure Loss Vector Clouds

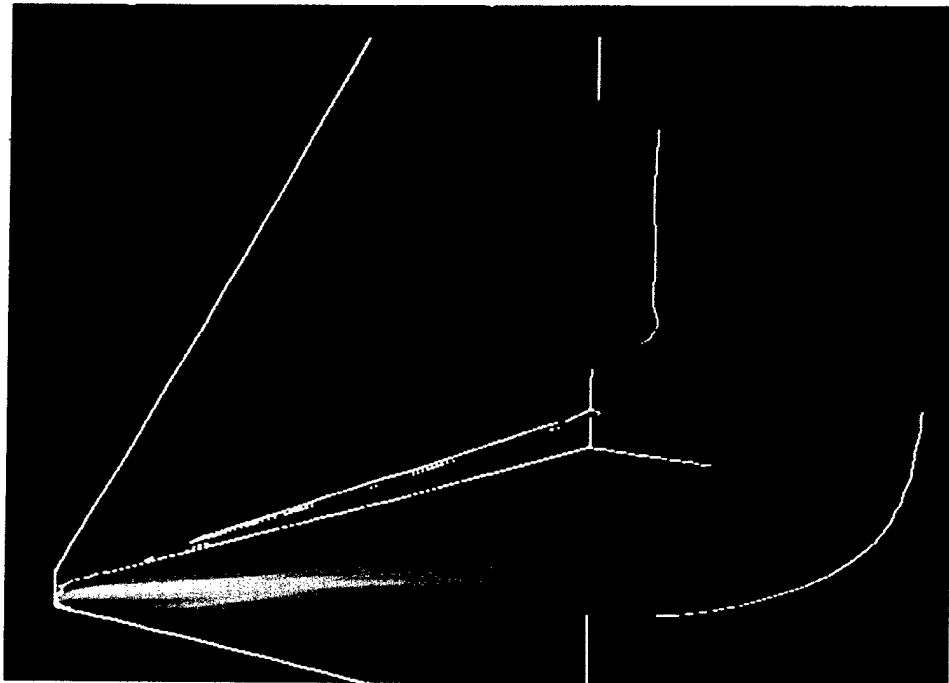Figure 2: Cutting plane example - full screen view
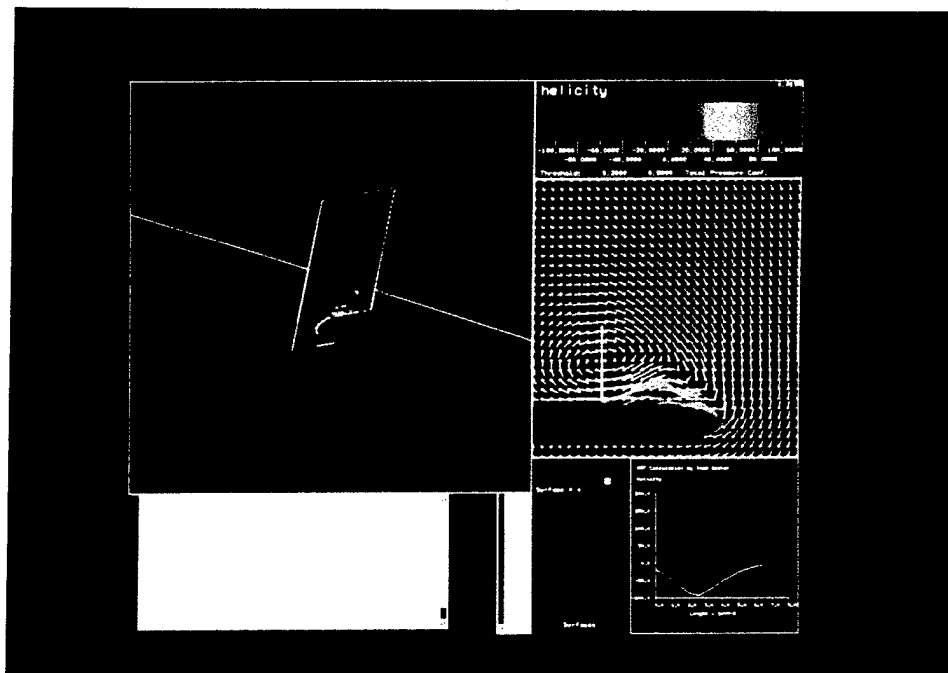
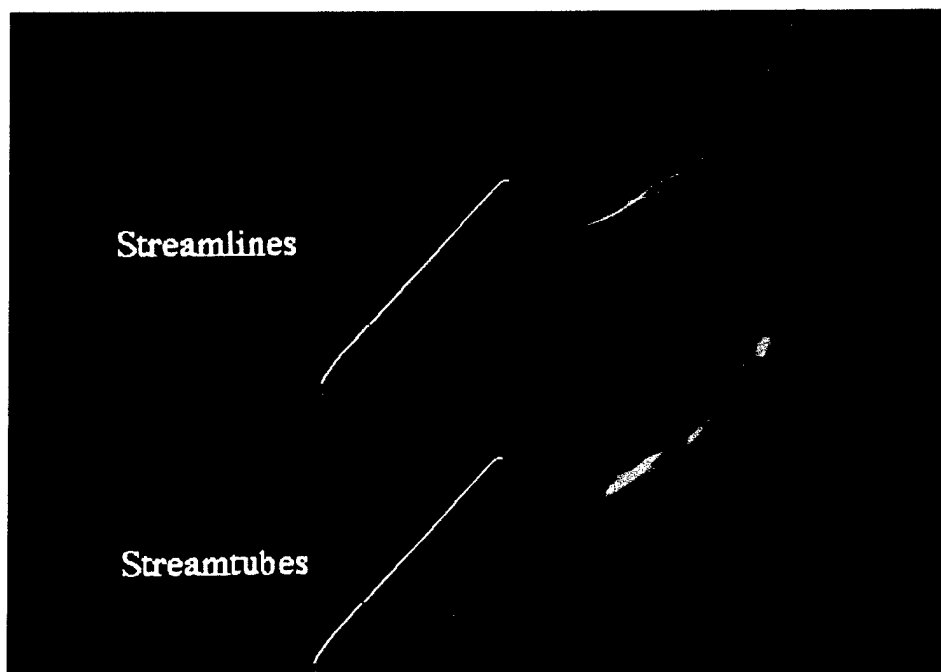Figure 3: Shock surfaces rendered by shock strength

Figure 4: Tufts example - full screen view

Figure 5: Comparison of streamlines and tubes